

AMENDMENTS TO THE DRAWINGS:

The formal drawing Replacement Sheet attached in connection with the above-identified application containing Figure 9 is being presented as new formal drawing sheet to be substituted for the previously submitted drawing sheet containing Figure 9. Figure 9 has been amended and the specific changes which have been made are listed below, whereby red-line versions of these figures are also being submitted herewith:

FIGURE 9:

The line of code containing "String.vatueOf" has been corrected to read "String.valueOf".

The line of code containing "DecordImage(pixData, PixDataMax);" has been corrected to read "DecodelImage(pixData, PixDataMax);".

FIG.9

RECEIVED
DEC 03 2003
Technology Center 2100

```
import java.awt.*;
import java.applet.*;
import java.net.*;
import java.io.*;
import java.awt.image.*;

public class view extends Applet
{
    Image img;

    /* init method. Called only once when applet is loaded. */
    public void init() {
        /* Acquire document number and page number from parameters.*/
        int docNumber=Integer.parseInt(getParameter("docNumber"));
        int pageNumber=Integer.parseInt(getParameter("pageNumber"));

        /* Produce file name /XXXX-YYYY-pix.data from document number and page number */
        String filename
        =String.valueOf(docNumber)+"-"+String.valueOf(pageNumber)+"-pix.data";

        /* Connect to facsimile apparatus and request for and acquire image data /XXXX-YYYY
        -pix.data */
        int pixData[ ]=new int [1728*2000/8];
        int pixDataMax=0;
        try{
            URL url=new URL(getCodeBase(),filename);
            URLConnection urlCon=url.openConnection();
            BufferedReader bufr =
                new bufferedReader(new InputStreamReader(urlCon.getInputStream()));
            int c; while ((c=bufr.read ())!=-1) pixData[pixDataMax++]=c;
            bufr.close();
        }catch(java.io.IOException ex){
            ex.printStackTrace();
        }

        /* Decode with known coding method to produce decoded image */
        DecodeImage(pixData, pixDataMax);
        img=createImage(new MemoryImage Source (1728,2000,pixData,0,1728))
    }

    /* paint method. Called from Java system when plotting is required. */
    public void paint(Graphics g){
        /* Display of decoded image */
        g.drewImago(img,0,0,this);
    }
}
```

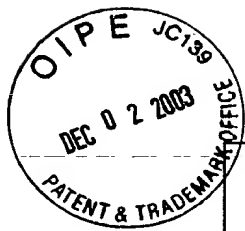


FIG.9



RECEIVED

DEC 03 2003

Technology Center 2100

```
import java.awt.*;
import java.applet.*;
import java.net.*;
import java.io.*;
import java.awt.image.*;
```

```
public class view extends Applet
{
    Image img;
```

```
/* init method. Called only once when applet is loaded. */
```

```
public void init() {
```

```
    /* Acquire document number and page number from parameters.*/
```

```
    int docNumber=Integer.parseInt(getParameter("docNumber"));
```

```
    int pageNumber=Integer.parseInt(getParameter("pageNumber"));
```

```
/* Produce file name /XXXX-YYYY-pix.data from document number and page number */
```

```
String filename
```

```
=String.valueOf(docNumber)+"-"+String.valuevalueOf(pageNumber)+"-"+pix.data";
```

```
/* Connect to facsimile apparatus and request for and acquire image data /XXXX-YYYY  
-pix.data */
```

```
int pixData[ ]=new int [1728*2000/8];
```

```
int pixDataMax=0;
```

```
try{
```

```
    URL url=new URL(getCodeBase(),filename);
```

```
    URLConnection urlCon=url.openConnection();
```

```
    BufferedcdReader bufr =
```

```
        new bufferedReader(new InputStreamReader(urlCon.getInputStream()));
```

```
    int c; while ((c=bufr.read ())!=-1) pixData[pixDataMax++]=c;
```

```
    bufr.close();
```

```
}catch(java.io.IOException ex){
```

```
    ex.printStrack Trace();
```

```
}
```

```
/* Decode with known coding method to produce decoded image */
```

Decode ~~Decord~~Image(pixData, pixDataMax);

```
img=createImage(new MemoryImage Source (1728,2000,pixData,0,1728))
```

```
}
```

```
/* paint method. Called from Java system when plotting is required. */
```

```
public void paint(Graphics g){
```

```
    /* Display of decoded image */
```

```
    g.drewImago(img,0,0,this);
```

```
}
```